Amendments to the Claims:

This listing of claims will replace all prior versions, and listings, of claims in the application:

Listing of Claims:

1. (currently amended) Method, performed by a suitably programmed processor, for software emulation of hard disks of a data processing platform at the level of ~~the~~ an operating system with parameterizable management ~~on the fly~~ of requests for writing and reading data, consisting ~~in~~ of:

creating ~~in a first step~~ a representation of a real hard disk, ~~in which the orders of~~ wherein the sequence and location for loading and execution of ~~certain~~ components of the operating system of ~~a~~ the data processing platform may be modified,

~~then in a second step~~ loading on said data processing platform one or more peripheral drivers, ~~among which~~ wherein at least one of the peripheral drivers ~~permits real dialogue~~ communicates with a data storage support containing the data of the ~~emulated~~ representation of the real hard disk, and

~~then~~ simulating ~~in a third step the~~ behavior of ~~a~~ the real hard disk for the operating system, wherein the method transforms programming contained on the real hard disk into an emulated hard disk capable of controlling read and write operations on a client station and among two or more client stations.

2. (currently amended) Method as claimed in claim 1, wherein the management of said data write requests that the operating system sends to the emulated hard disk is accomplished at the peripheral driver level and/or at the level of an optional hard disk server service on ~~the~~ a data processing network, the written data ~~bein~~ being stored according to the parameterization of said peripheral drivers and/or said service server of the hard disk on the network

- either directly in the support containing the emulated hard disk,
- or in the memory, random access or virtual, accessible to the operating system using the emulated hard disk,

- or in a volatile storage space accessible to the operating system using the emulated hard disk,
- or in a non-volatile storage space accessible to the operating system using the emulated hard disk,
- or in a volatile storage space accessible to the server service of emulated hard disks on a data processing network~~.~~,
- or in a non-volatile storage space accessible to the server service of emulated hard disks on a data processing network.

3. (currently amended) Method as claimed in claim 1, wherein the management of the data reading requests that the operating system issues to the emulated hard disk is accomplished at the peripheral driver level and/or at the level of an optional hard disk server service on ~~the~~ a data processing network, the readings of previously written data being performed in the storage space:

- either directly in the support containing the emulated hard disk,
- or in the random access or virtual memory accessible to the operating system using the emulated hard disk,
- or in a volatile storage space accessible to the operating system using the emulated hard disk,
- or in a nonvolatile storage space accessible to the operating system using the emulated hard disk,
- or in a volatile storage space accessible to the server service of emulated hard disks on a data processing network,
- or in a non-volatile storage space accessible to the server service of emulated hard disks on a data processing network.

4. (currently amended) Method as claimed in claim 1, wherein the emulation of the hard disk provided to the operating system of ~~a~~ the client station is accomplished by the agency of a single, monolithic peripheral driver which communicates with the operating system in the manner of a hard disk and which communicates with the support containing the data of said emulated hard disk in a manner specific to this support.

5. (previously presented) Method as claimed in claim 1, wherein the data of the emulated hard disk or disks are accessible to the client stations via a data processing network.

6. (currently amended) Method as claimed in claim 1, wherein ~~if~~ when an emulated hard disk is ~~to be~~ started up, the method further comprises ~~a low level micro-software module is responsible for access to the data contained in said emulated hard disk by~~ providing ~~an~~ a low level micro-software module to access the data contained in the emulated hard disk, wherein ~~interface of the type of that provided by the micro-software having access to the data of real hard disks by~~ the operating system is started up at the client station.

7. (currently amended) Method as claimed in claim 6, wherein the data processing network comprises a plurality of client stations, the client stations ~~wherein the micro-software could, in the case of computers~~ using ~~bootup memory programs of the “PXE” type (PXE~~ a bootup PROM~~)~~, the method further comprising using the bootup PROM to control~~use the functions made available by these “PROMS” for controlling~~ communications via the data processing network ~~independently of the network interface model employed~~.

8. (currently amended) Method as claimed in claim 7, wherein the low level micro-software module is loaded in the memory of the client station and executed ~~by~~ using ~~the functions made available by~~ the bootup ~~“~~PROM~~”~~.

9. (currently amended) Method as claimed in claim 6, wherein the low level micro-software module is loaded in ~~the~~ memory of the client station and executed as a component of the ~~basic standard micro-software (~~BIOS~~, for example)~~ of the client station, said low level micro-software module providing the same functions as the access services on real hard disks ~~normally~~ provided by the ~~basic standard micro- software~~ BIOS.

10. (currently amended) Method as claimed in claim 6, wherein the low-level micro-software is loaded in ~~the~~ memory of the client station from a third party data support supported as a startup peripheral by the client station.

11. (previously presented) Method as claimed in claim 5, wherein at least one peripheral driver loaded and executed by the operating system of the client station

provides the functions of access, via the data processing network, to the data contained in the emulated hard disks.

12. (currently amended) Method as claimed in claim 1, wherein if the data support containing the data of the emulated hard disk(s) is a support that does not ~~support~~ provide for writing in real time, or ~~the system of hard disk emulation according to the invention is parameterized~~ does not ~~to~~ accept ~~the~~ writing of data directly in the support containing the data of the emulated hard disk, ~~the peripheral drivers providing the emulation of the hard disk at the client stations method~~ the data writing requests issued by the operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in a storage space different from the data support containing the data of the emulated hard disk(s)

13. (previously presented) Method as claimed in claim 12, wherein the data writing requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the random access memory of the client station.

14. (previously presented) Method as claimed in claim 12, wherein the data writing requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the virtual memory of the client station.

15. (previously presented) Method as claimed in claim 12, wherein the data writing requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in a data file accessible to the operating system of the client station.

16. (currently amended) Method as claimed in claim 1, wherein the data writing requests issued by the operating system to the emulated hard disk(s) are~~, at a given moment,~~ redirected to ~~one and only one~~ a single storage space~~;~~, and wherein the storage space in which the written data are redirected may be changed ~~on the fly~~ during an operating session of the operating system of a client station.

17. (currently amended) Method as claimed in claim 12, wherein the storage space used for storage of the written data may be volatile~~, i.e. be emptied of data that are~~

~~stored in each new operating session of the client station operating system~~ or nonvolatile so as to permit the written data of an operating session of the operating system to persist from one client station to another.

18. (currently amended) Method as claimed in claim 16, wherein the volatile character of the redirections of the written data is determined upon initialization of the operating session of the operating system of ~~a~~ the client station.

19. (currently amended) Method as claimed in claim 1, wherein the data reading requests issued by the operating system ~~may be~~ are performed in different storage spaces during an operating session of the operating system of a client station.

20. (currently amended) Method as claimed in claim 19, wherein the data reading requests issued by the operating system to an emulated hard disk carried out in different storage spaces ~~are following~~ follow an order of priority.

21. (currently amended) Method as claimed in claim 5, wherein a ~~specific program called "server software"~~ server program is in charge at one of the client stations of the data processing network, on the one hand, of the communications via the data processing network with the client stations accessing the emulated hard disks, and on the other, of accessing the data support containing the data of the emulated hard disks.

22. (currently amended) Method as claimed in claim 21, wherein if the hard disk emulation system is parameterized so that the data write requests received by the server ~~software~~ program are intended for a specific emulated hard disk they are not redirected but stored directly in a support containing the data of the emulated hard disk itself, and only one client station can access said emulated hard disk at a given time.

23. (currently amended) Method as claimed in claim 21, wherein in order to permit several client stations to access an emulated hard disk simultaneously, the server ~~software~~ program is capable of redirecting specifically the data write requests issued by a client station A to a given storage space, and of redirecting the data write requests issued by another client station B to another given storage space.

24. (currently amended) Method as claimed in claim 1, wherein in order to permit the startup from and/or simultaneous access to the same emulated hard disk or

100% identical copies of the same emulated hard disk, ~~certain constituent~~ components of the operating system ~~invention~~ loaded and executed by the client stations or server ~~software~~ program are capable of modifying, ~~on the fly~~ during or before their ~~effective~~ use by the operating system, ~~of certain~~ data contained in the emulated hard disk.

25. (currently amended) Method as claimed in claim 1, wherein the emulation ~~itself~~ is ~~perfomed~~ performed for the operating system of the client stations at ~~the~~ a level of ~~the~~ a class of virtual peripherals of ~~the~~ a file system type ~~as in the products “Qualystem LiteNET PC 1.x” and “Qualystem LAN PC 2.x (CIFS or SMB file system) or “Qualystem Rescue 1.x” (1SO9660/Joliet, CDFS or UDF file System)~~.

26. (currently amended) Method as claimed in claim 1, wherein the emulation is ~~perfomed~~ performed for the operating system of the client stations at the level of the class of disk peripherals itself and not at the file system level.

27. (currently amended) Method as claimed in claim 1, wherein ~~the significant~~ data contained in the emulated hard disk ~~and~~ are copied by a software tool executed at a ~~references~~ client station from ~~a~~ the real hard disk ~~called the reference hard disk that is accessible to the operating system of said reference station~~.

28. (previously presented) Method as claimed in claim 27, wherein the software tool creates an image directory that contains the data of the emulated hard disk.

29. (previously presented) Method as claimed in claim 27, wherein the software tool creates an image file that contains the data of the emulated hard disk.

30. (currently amended) Method as claimed in claim 1, wherein in order to permit startup from an emulated hard disk, the sequence of loading of the components of the operating system ~~requires an adjustment~~ is modified so that all components of the operating system on which the peripheral drivers permitting access to the emulated hard disk ~~according to the invention~~ depend are loaded and usable at the moment when the operating system ~~needs to access~~ accesses the emulated hard disk by using the peripheral drivers ~~and no longer by using the firmware functions (BIOS)~~.

31. (currently amended) Method as claimed in claim 21, wherein in order to accelerate the simultaneous access by several client stations to the same emulated hard

disk whose data are contained in a data support accessible to ~~the~~ a server station, the data are sent by the server station to the client stations ~~within the scope of the hard disk emulation globally and at a single time by~~ using ~~the~~ "broadcast" or "multicast" mechanisms~~ instead of the "unicast" mechanism~~.

32. (currently amended) Method as claimed in claim 31, wherein the data sent by "broadcast" or by "multicast" by the server station are stored by the client stations that accept them in a local cache situated in the memory~~ (real or virtual)~~ of said client stations.

33. (currently amended) Method as claimed in claim 31, wherein a ~~reading~~ read request for data in the emulated hard disk issued by the operating system of a client station generates an ~~explicit~~ data reading request sent to the server station only if said data are not already present in said local cache.

34. (previously presented) Method as claimed in claim 33, wherein the data read in the local cache are removed after being read by the client station so as to free up space in said local cache.

35. (currently amended) Method as claimed in claim 31, wherein the decision to send~~, within the scope of the hard disk emulation according to the invention, of~~ data by "multicast/broadcast" ~~or unicast"~~ is made at the server module level which provides the functionalities necessary for the hard disk emulation at the client stations.

36. (currently amended) Method as claimed in claim 31, wherein the client stations may modify their subscription to receiving the data sent via "broadcast/multicast" by the server station~~ within the scope of emulation of hard disks according to the invention~~.

37. (previously presented)Method as claimed in claim 32, wherein the client stations may erase the data from the local cache after a certain parameterizable time.

38. (currently amended) Method as claimed in claim 5, wherein data contained in the emulated the server module making the hard disks available to client stations may use any suitable network protocol including one of NVD, iSCSI, and SMB/CFIS.

39. (currently amended) Method as claimed in claim 5, wherein ~~the~~ a low level software program executed by the client stations ~~and permitting~~ permits access to the data contained in the emulated hard disks ~~may use~~ using any suitable network protocol including one of NVD, iSCSI, and SMB/CFIS.

40. (currently amended) Method as claimed in claim 11, wherein the peripheral driver(s) ~~according to the invention~~ executed by the client stations ~~and permitting~~ permit access to the data contained in the emulated hard disks ~~may use~~ any suitable network protocol including one of NVD, iSCSI, and SMB/CFIS.

41. (currently amended) Method as claimed in claim 21, wherein if the data support containing the data of the emulated hard disk(s) is a support that does not ~~support~~provide writing in real time, or ~~the system of hard disk emulation according to the invention is parameterized~~ does not ~~to~~ accept ~~the~~ write operations directly in the support containing the data of the emulated hard disk, the server ~~software~~ program providing the emulation of the hard disk at the client stations processes the data write requests issued by the operating system to the emulated hard disk(s) in such a way that the written data are stored in a storage space different from the data support containing the data of the emulated hard disk(s)

42. (previously presented) Method as claimed in claim 21, wherein the data write requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the random access memory of the server station.

43. (previously presented) Method as claimed in claim 21, wherein the data write requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in the virtual memory of the server station.

44. (previously presented) Method as claimed in claim 21, wherein the data write requests issued by the client station operating system to the emulated hard disk(s) are processed in such a way that the written data are stored in a data tile accessible to the server software.

45. (currently amended) Method as claimed in claim 21, wherein the storage space used for storage of the written data may be volatile~~, i.e. be emptied of data that are stored in each new operating session of the client station operating system~~ or nonvolatile so as to permit the written data of an operating session of the operating system to persist from one client station to another.

46. (currently amended) Method as claimed in claim 16, wherein the volatile character of the redirections of the written data is determined upon initialization of the operating session of the operating system of ~~a~~ the client station.